University of Niagara Falls Canada

CPSC-500 - 2

SQL Database

Assignment 2

Group -3

Dhruv Patel	**|**	**NF1000200**
Nidhi Tailor	**|**	**NF1018012**
Yuvraj Singh	**|**	**NF1016125**

Professor

Sundus Shanef

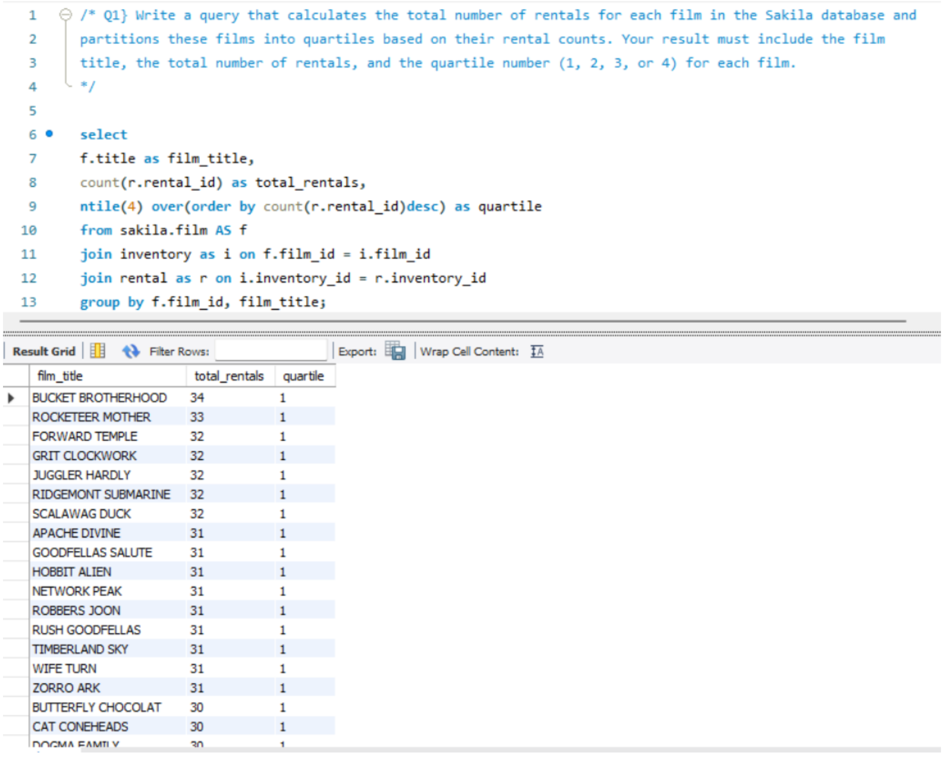
March 9th, 2025

Question 1

Write a query that calculates the total number of rentals for each film in the Sakila database and partitions these films into quartiles based on their rental counts. Your result must include the film title, the total number of rentals, and the quartile number (1, 2, 3, or 4) for each film.

Ans:

```
SELECT

        f.title AS film_title,

COUNT(r.rental_id) AS total_rentals,

NTILE (4) OVER (ORDER BY COUNT(r.rental_id)DESC) AS quartile

FROM sakila.film AS f

JOIN inventory AS i ON f.film_id = i.film_id

JOIN rental AS r ON i.inventory_id = r.inventory_id

GROUP BY f.film_id, film_title;
```



```
/* Q1} Write a query that calculates the total number of rentals for each film in the Sakila database and
partitions these films into quartiles based on their rental counts. Your result must include the film
title, the total number of rentals, and the quartile number (1, 2, 3, or 4) for each film.
*/

select
f.title as film_title,
count(r.rental_id) as total_rentals,
ntile(4) over(order by count(r.rental_id)desc) as quartile
from sakila.film AS f
join inventory as i on f.film_id = i.film_id
join rental as r on i.inventory_id = r.inventory_id
group by f.film_id, film_title;
```

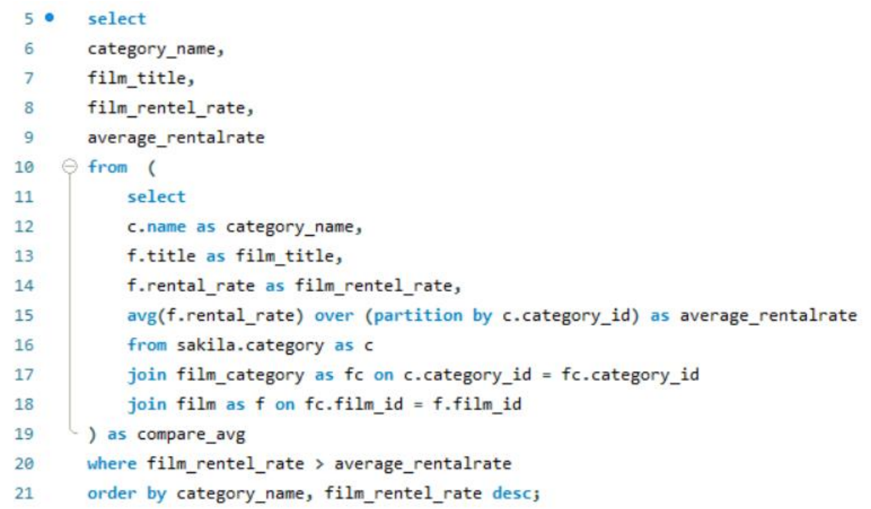
film_title	total_rentals	quartile
BUCKET BROTHERHOOD	34	1
ROCKETEER MOTHER	33	1
FORWARD TEMPLE	32	1
GRIT CLOCKWORK	32	1
JUGGLER HARDLY	32	1
RIDGEMONT SUBMARINE	32	1
SCALAWAG DUCK	32	1
APACHE DIVINE	31	1
GOODFELLAS SALUTE	31	1
HOBBIT ALIEN	31	1
NETWORK PEAK	31	1
ROBBERS JOON	31	1
RUSH GOODFELLAS	31	1
TIMBERLAND SKY	31	1
WIFE TURN	31	1
ZORRO ARK	31	1
BUTTERFLY CHOCOLAT	30	1
CAT CONEHEADS	30	1
DOGMA FAMILY	30	1

Question 2

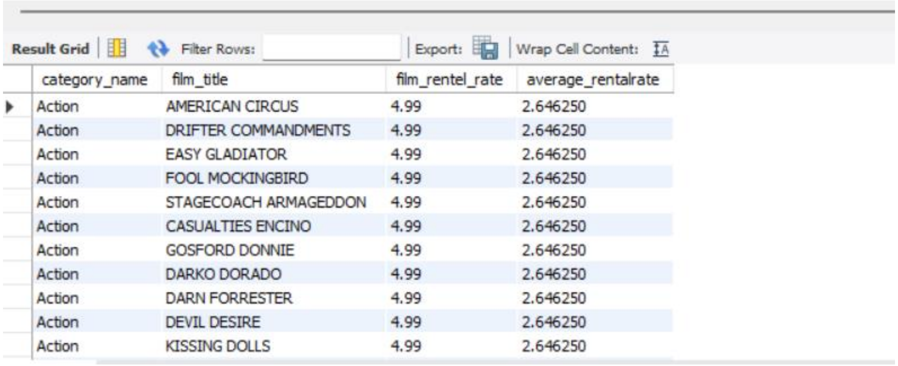
For each film category, write a query to identify the films with rental rates that exceed the average rental rate for that category. The output should display the film category name, film title, the film’s rental rate, and the average rental rate computed for the category.

Ans:

```
SELECT
category_name,
film_title,
film_rentel_rate,
average_rentalrate
FROM (
        SELECT
        c.name AS category_name,
        f.title AS film_title,
        f.rental_rate AS film_rentel_rate,
        AVG(f.rental_rate) OVER (PARTITION BY c.category_id) AS average_rentalrate
        FROM sakila.category AS c
        JOIN film_category AS fc ON c.category_id = fc.category_id
        JOIN film AS f ON fc.film_id = f.film_id
) AS compare_avg
WHERE film_rentel_rate > average_rentalrate
ORDER BY category_name, film_rentel_rate DESC;
```



```sql
select
category_name,
film_title,
film_rentel_rate,
average_rentalrate
from  (
    select
    c.name as category_name,
    f.title as film_title,
    f.rental_rate as film_rentel_rate,
    avg(f.rental_rate) over (partition by c.category_id) as average_rentalrate
    from sakila.category as c
    join film_category as fc on c.category_id = fc.category_id
    join film as f on fc.film_id = f.film_id
) as compare_avg
where film_rentel_rate > average_rentalrate
order by category_name, film_rentel_rate desc;
```



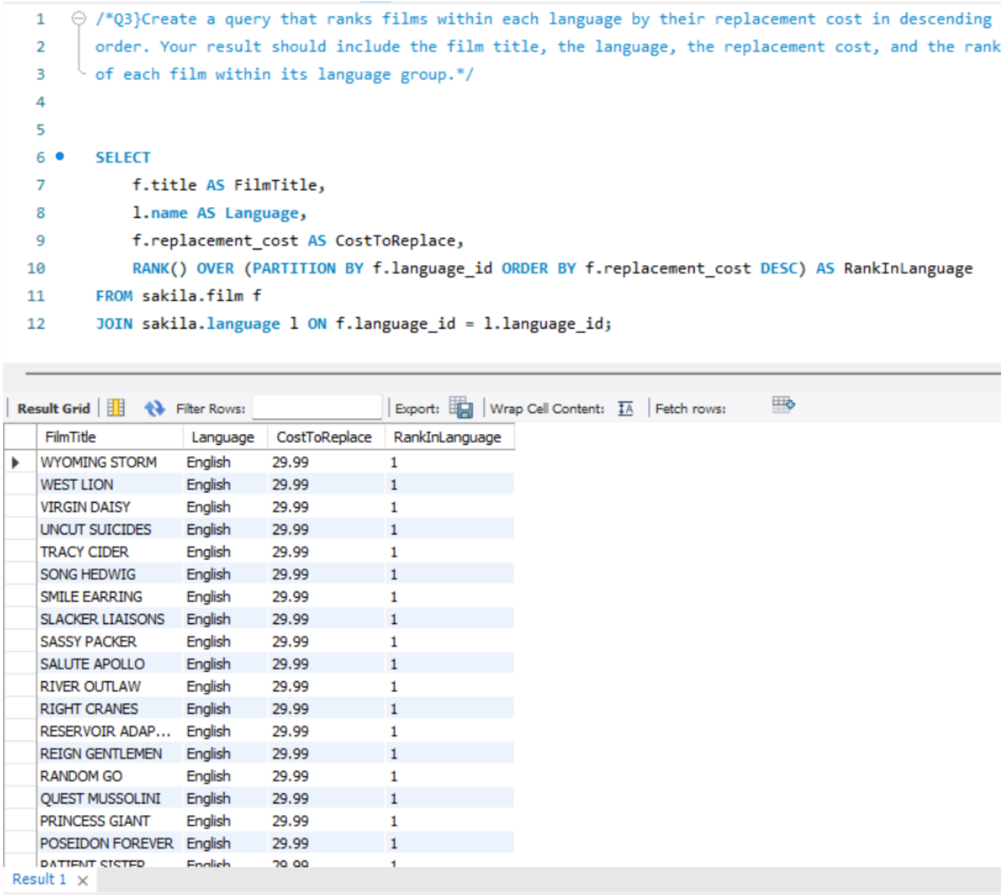
category_name	film_title	film_rentel_rate	average_rentalrate
Action	AMERICAN CIRCUS	4.99	2.646250
Action	DRIFTER COMMANDMENTS	4.99	2.646250
Action	EASY GLADIATOR	4.99	2.646250
Action	FOOL MOCKINGBIRD	4.99	2.646250
Action	STAGECOACH ARMAGEDDON	4.99	2.646250
Action	CASUALTIES ENCINO	4.99	2.646250
Action	GOSFORD DONNIE	4.99	2.646250
Action	DARKO DORADO	4.99	2.646250
Action	DARN FORRESTER	4.99	2.646250
Action	DEVIL DESIRE	4.99	2.646250
Action	KISSING DOLLS	4.99	2.646250

Question 3

Create a query that ranks films within each language by their replacement cost in descending order. Your result should include the film title, the language, the replacement cost, and the rank of each film within its language group.

Ans:

```
SELECT

    f.title AS FilmTitle,

    l.name AS Language,

    f.replacement_cost AS CostToReplace,

RANK() OVER (PARTITION BY f.language_id ORDER BY f.replacement_cost DESC)
AS RankInLanguage

FROM sakila.film f

JOIN sakila.language l ON f.language_id = l.language_id;
```



```sql
/*Q3}Create a query that ranks films within each language by their replacement cost in descending
order. Your result should include the film title, the language, the replacement cost, and the rank
of each film within its language group.*/


SELECT
    f.title AS FilmTitle,
    l.name AS Language,
    f.replacement_cost AS CostToReplace,
    RANK() OVER (PARTITION BY f.language_id ORDER BY f.replacement_cost DESC) AS RankInLanguage
FROM sakila.film f
JOIN sakila.language l ON f.language_id = l.language_id;
```

FilmTitle	Language	CostToReplace	RankInLanguage
WYOMING STORM	English	29.99	1
WEST LION	English	29.99	1
VIRGIN DAISY	English	29.99	1
UNCUT SUICIDES	English	29.99	1
TRACY CIDER	English	29.99	1
SONG HEDWIG	English	29.99	1
SMILE EARRING	English	29.99	1
SLACKER LIAISONS	English	29.99	1
SASSY PACKER	English	29.99	1
SALUTE APOLLO	English	29.99	1
RIVER OUTLAW	English	29.99	1
RIGHT CRANES	English	29.99	1
RESERVOIR ADAP...	English	29.99	1
REIGN GENTLEMEN	English	29.99	1
RANDOM GO	English	29.99	1
QUEST MUSSOLINI	English	29.99	1
PRINCESS GIANT	English	29.99	1
POSEIDON FOREVER	English	29.99	1
PATIENT SISTER	English	29.99	1

Question 4

Develop a query to compute the interval, in days, between consecutive rentals for each customer in the Sakila database. The output should look like the followings:

rental_id	customer_id	rental_date	days_until_next_rental
76	1	2005-05-25 11:30:37	3
573	1	2005-05-28 10:35:23	18
1185	1	2005-06-15 00:54:12	0
1422	1	2005-06-15 18:02:53	0
1476	1	2005-06-15 21:08:46	1
….	…	…	…

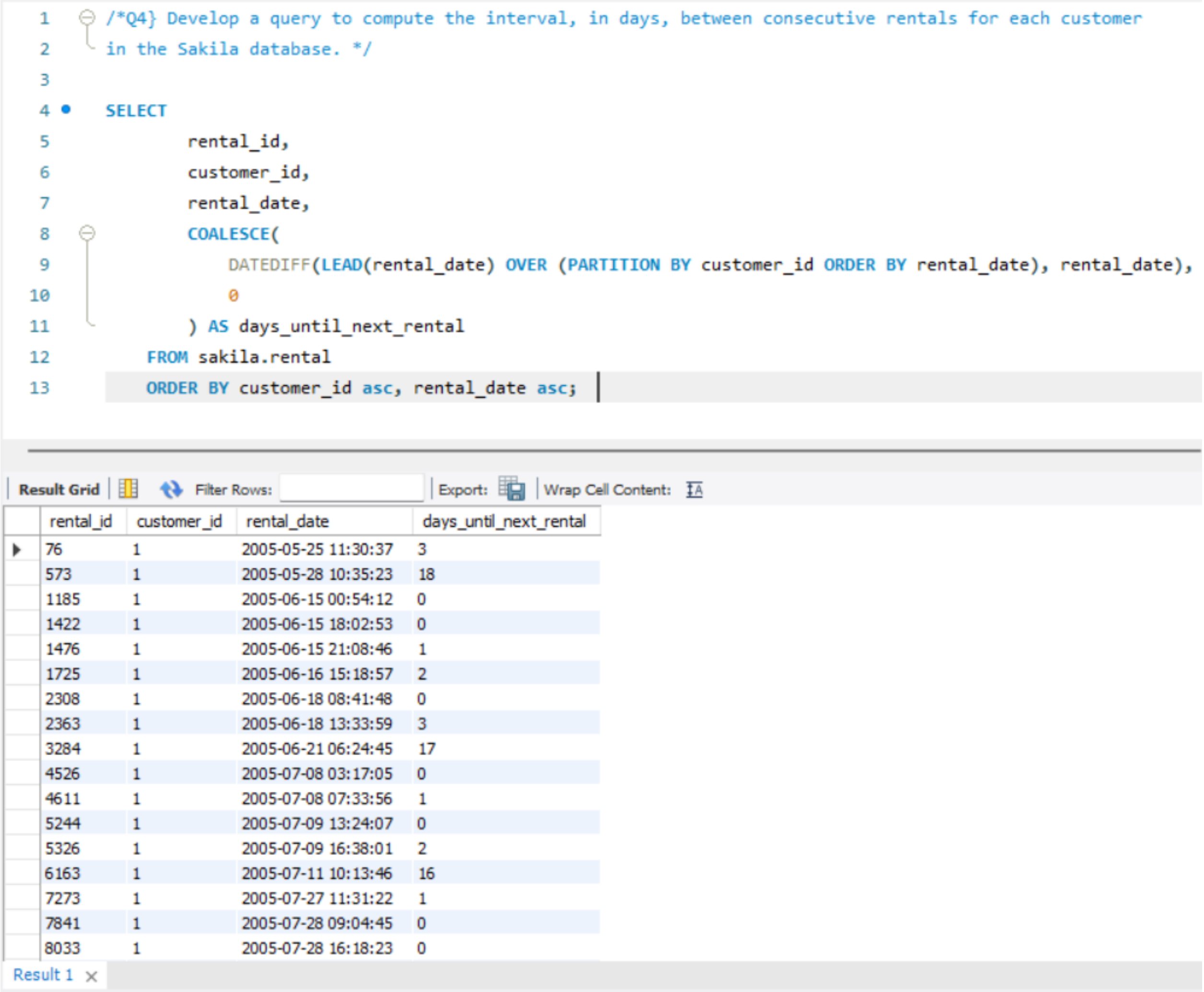
Ans:

```
SELECT
    rental_id,
    customer_id,
    rental_date,
DATEDIFF(
    LEAD(r.rental_date) OVER (PARTITION BY r.customer_id ORDER BY
    r.rental_date),
    r.rental_date
```

) AS days_until_next_rental

FROM sakila.rental AS r

ORDER BY customer_id asc, rental_date asc;



```
/*Q4} Develop a query to compute the interval, in days, between consecutive rentals for each customer
in the Sakila database. */

SELECT
        rental_id,
        customer_id,
        rental_date,
        COALESCE(
            DATEDIFF(LEAD(rental_date) OVER (PARTITION BY customer_id ORDER BY rental_date), rental_date),
            0
        ) AS days_until_next_rental
    FROM sakila.rental
    ORDER BY customer_id asc, rental_date asc;
```

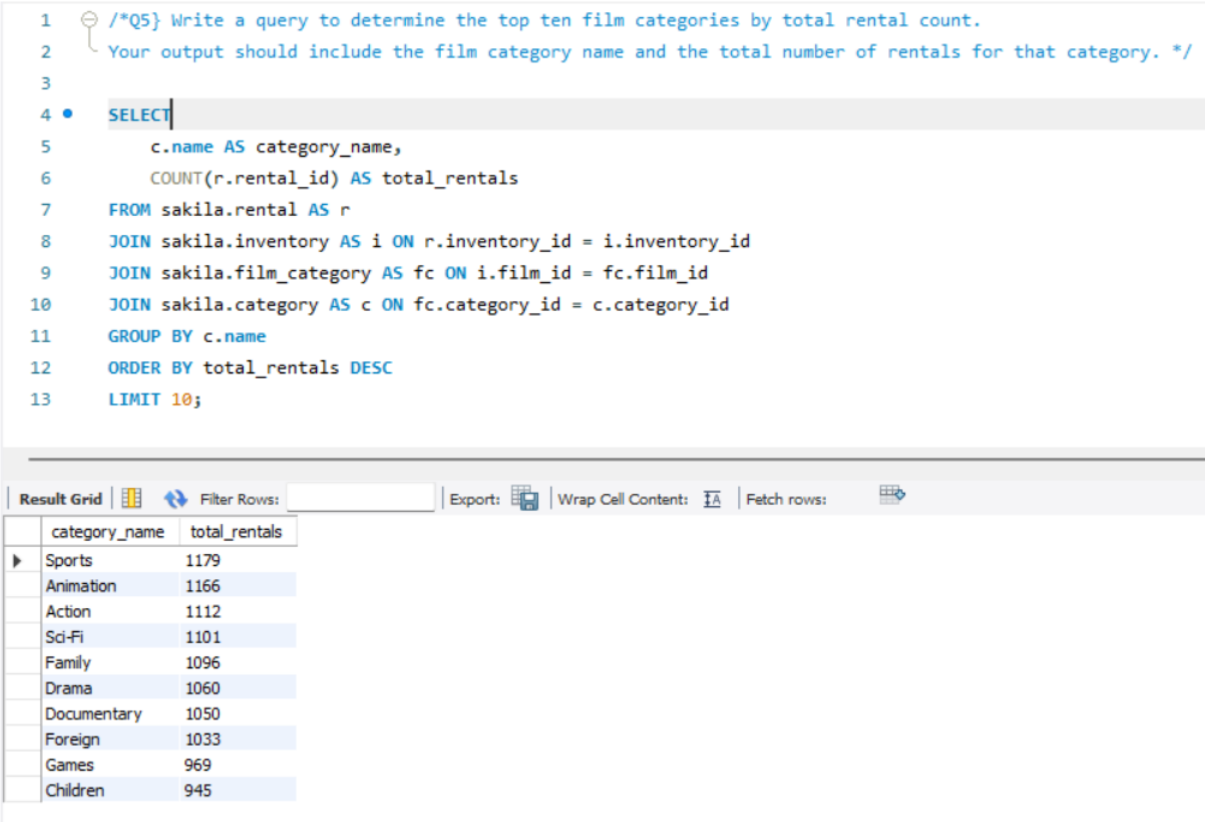
rental_id	customer_id	rental_date	days_until_next_rental
76	1	2005-05-25 11:30:37	3
573	1	2005-05-28 10:35:23	18
1185	1	2005-06-15 00:54:12	0
1422	1	2005-06-15 18:02:53	0
1476	1	2005-06-15 21:08:46	1
1725	1	2005-06-16 15:18:57	2
2308	1	2005-06-18 08:41:48	0
2363	1	2005-06-18 13:33:59	3
3284	1	2005-06-21 06:24:45	17
4526	1	2005-07-08 03:17:05	0
4611	1	2005-07-08 07:33:56	1
5244	1	2005-07-09 13:24:07	0
5326	1	2005-07-09 16:38:01	2
6163	1	2005-07-11 10:13:46	16
7273	1	2005-07-27 11:31:22	1
7841	1	2005-07-28 09:04:45	0
8033	1	2005-07-28 16:18:23	0

Question 5

Write a query to determine the top ten film categories by total rental count. Your output should include the film category name and the total number of rentals for that category.

Ans:

```
SELECT
        c.name AS category_name,
        COUNT(r.rental_id) AS total_rentals
FROM sakila.rental AS r
JOIN sakila.inventory AS i ON r.inventory_id = i.inventory_id
JOIN sakila.film_category AS fc ON i.film_id = fc.film_id
JOIN sakila.category AS c ON fc.category_id = c.category_id
GROUP BY c.name
ORDER BY total_rentals DESC
LIMIT 10;
```



```
/*Q5} Write a query to determine the top ten film categories by total rental count.
Your output should include the film category name and the total number of rentals for that category. */

SELECT
    c.name AS category_name,
    COUNT(r.rental_id) AS total_rentals
FROM sakila.rental AS r
JOIN sakila.inventory AS i ON r.inventory_id = i.inventory_id
JOIN sakila.film_category AS fc ON i.film_id = fc.film_id
JOIN sakila.category AS c ON fc.category_id = c.category_id
GROUP BY c.name
ORDER BY total_rentals DESC
LIMIT 10;
```

category_name	total_rentals
Sports	1179
Animation	1166
Action	1112
Sci-Fi	1101
Family	1096
Drama	1060
Documentary	1050
Foreign	1033
Games	969
Children	945